Data Validation Tool

# Introduction:

This document provides details of custom tool written in Python that was used for Data Validation requirements of our client’s Database Migration project from Netezza to SQL Server.

# About the Tool:

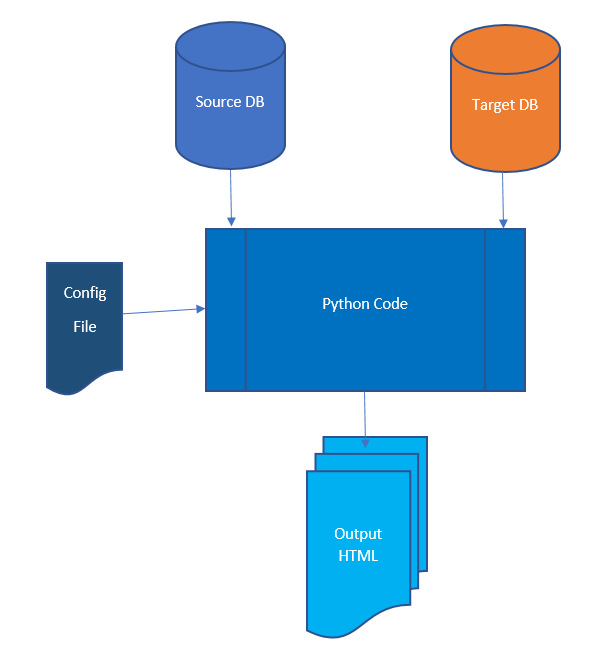


Figure 1: Validation Tool High Level Overview

The tool validates the following for Tables/Views/Queries run against Source and Target DB

1. Total row count of Table/View/Query Output
2. Metadata of the Table Definition
3. Data Values of each field available in either of the Table

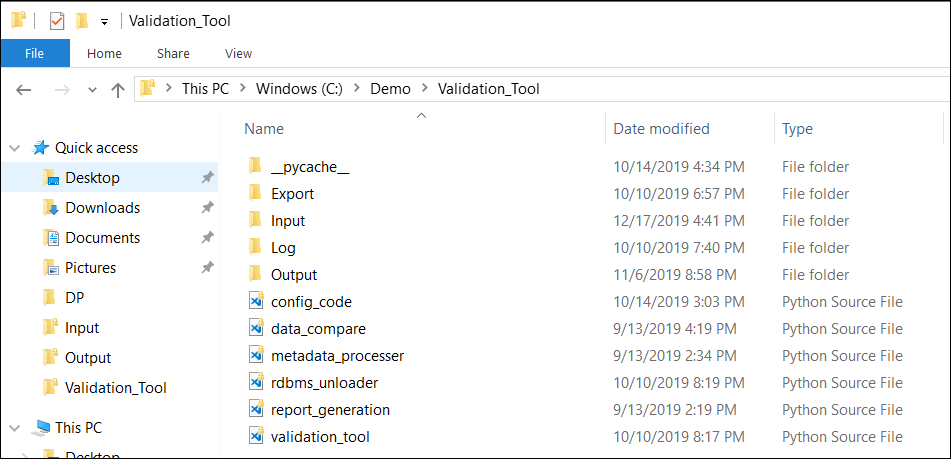
This Data Validation tool is a refactored version of a tool, originally developed as part of a MongoDB Hackathon by

* Peter Youngren Moses,
* Saravana Sekar
* Shreyashi Banik
* Shreeja Hm

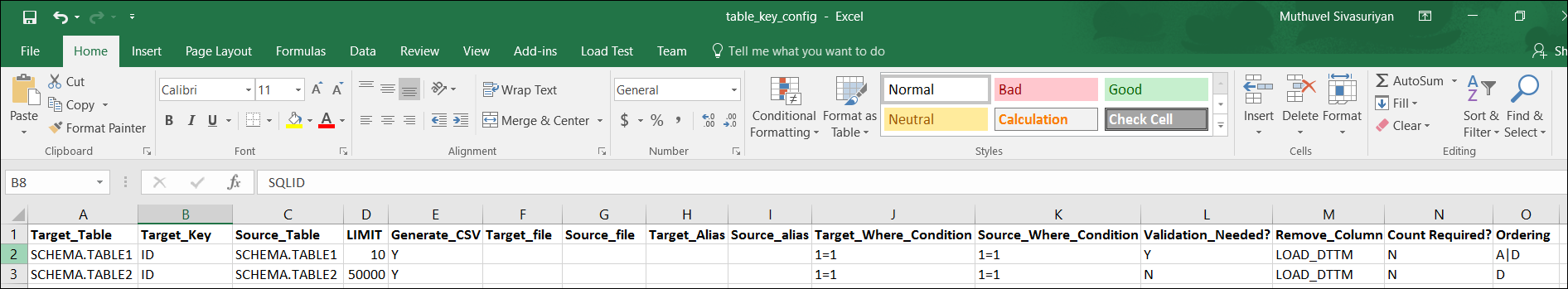
(ref: <https://codehub-repo.ad.infosys.com/Peter.Moses/mongo_bot_project.git> ).

# Prerequisites:

1. Database details and credentials to be updated in Config\_code.py
2. Table Names/Query for which data needs to be compared and validated
3. Python installed and setup up in local machine with required libraries
4. Space Available in local machine for various file outputs
5. The code placed in appropriate path as required(e.g. C:/Demo/Validation\_tool)

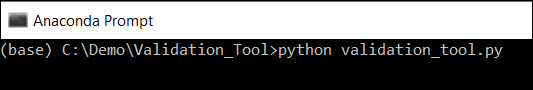


1. Table Key Config file setup with required information as per the requirement



## Steps to Run:

Run the code at python prompt as below



## Table Key Configuration file:

Table Key configuration is a csv file used to configure the parameters and initial settings required for the validation tool. The following are the parameters that are configurable:

|  |  |
| --- | --- |
| **Target\_Table** | **Name of the Target Table** |
| Target Key | Key Column(s) which identifies the records in the Source and Target table uniquely |
| Source\_Table | Name of the Source Table |
| LIMIT | Number of records to be retrieved for record validation |
| Generate CSV | Whether Data has to be exported to a csv file or not |
| Target\_File | Path of the Target Query, if any |
| Source File | Path of the Source Query, if any |
| Target Alias | Target Column Alias, if any, to be entered |
| Source Alias | Source Column Alias, if any, to be entered |
| Target\_Where\_Condition | Target Table constraints, if any, to be entered |
| Source\_Where\_Condition | Source Table constraints, if any, to be entered |
| Validation\_Needed? | Flag to be set to ‘Y’ if the table is required to be taken up for validation |
| Remove Columns | List of Columns that can be ignored by the program for validation |
| Count Required? | Flag to be set to ‘Y’ if Count check is to be performed |
| Ordering | Whether the ordering should be in Ascending or Descending order |

# Output Data Validation Report:

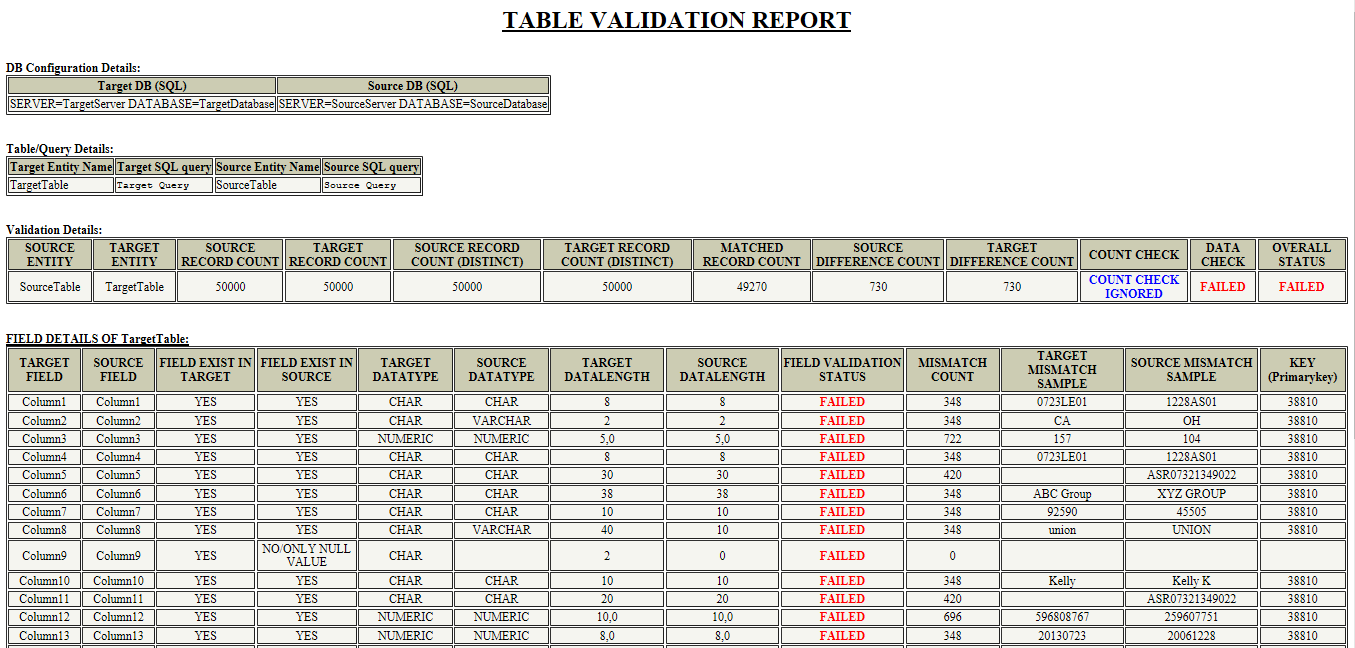


Figure 2: Sample Screenshot of Data Validation report

The below table contains the counts displayed under validation details in the output html report:

|  |  |
| --- | --- |
| **Count Type** | **Description** |
| Source Count | Total number of rows in source table/query |
| Target Count | Total number of rows in target table/query |
| Source Record Count | Number of rows in Source result set (<=configured for source as LIMIT) |
| Target Record Count | Number of rows in Target result set (<=configured for Target as LIMIT) |
| Source Record Count(Distinct) | Numbers of rows in source table/query result which are distinct based on the Key |
| Target Record Count(Distinct) | Numbers of rows in target table/query result which are distinct Key |

Additional details available under validation details in the output html report:

|  |  |  |
| --- | --- | --- |
| **Field** | **Status** | **Description** |
| Count Check | COUNT MATCHED | If Source Count= Target Count |
| DIFFERENCE IN COUNT | If Source Count <> Target Count |
| Data Check | SUCCESS | Sample rows in source matches with sample rows in target |
| FAILED | Sample rows in source does not matches with sample rows in target |
| Overall Status | SUCCESS | if both Count Check and Data Check gets ‘SUCCESS’ |
| FAILED | if any one Count Check or Data Check gets ‘FAILED’ |

# Field Level Validation Detail:

The field details section of the output html file contains the following information:

|  |  |  |
| --- | --- | --- |
| **Field** | **Status** | **Reason** |
| **Target/Source Field** | E.g. TYPE | Field Name having issue of Target/Source respectively |
| **Field Exist in Target/Source** | YES | If the field exists in Target/Source table respectively |
| No/only NULL value | If the field is all NULL or does not exists in Target/Source table respectively |
| **Target/Source Inferred Datatype** | E.g.: CHAR,VARCHAR | the DDL Defined Data Type for Target and Source respectively |
| **Target/Source Inferred Data length** | E.g. 1,9 | the DDL Defined Data length for Target and Source respectively |
| **Field Validation Status** | FAILED | Displayed as FAILED if the field has failed either for   * Data Validation (or) * Metadata Validation |
| **Mismatch Count** | E.g. 58 | No of records having mismatch in that field for the rows retrieved |
| **Target/source Mismatch Sample** | E.g. Target:670.38, Source:664.44 | Data value of the field in Target/Source |
| **KEY (Key1, key2)** | E.g.15981\_90 | KEY column values for that field mismatch for the retrieved sample are separated by underscore. |

Apart from the HTML Report, for any further analysis, the validation Source and Target data are stored as .csv files, Query execution time details are stored in the log file for performance check.

The result and issue files(superset of HTML files)are also consolidated into individual .csv files. Query execution time details are stored in the log file for performance check.

# Technical Details:

The steps involved in the working of the tool are as follows –

* The Table names in the configuration file are read iteratively.
* For each Table configured to be ‘Y’ in “Validation\_Needed?” column, the following process takes place:
* The query to be used is generated during the run time based on the table and other config file details like Row Limit to be applied, columns to be ignored. Exact Query for the source/target Query as defined by user is used only in case the required full file path for the file containing the same is provided in the configuration file columns – “Target\_file”/”Source\_file”
* Source and Target data is extracted and loaded in to a dictionary in a key-value pair format with key column values separated by underscore as Keys and all the available columns as Value.
* Metadata is generated for both Source and Target - Data Type, NotNull and Size based on the column-level table Information Schema
* Comparison is done for every row extracted, based on Matching Keys for each of the available columns (common in both Source and Target).
* In the event of a mismatch in any given column(field) in a table, the column is marked as having issue along with the number of records with similar mismatch and a sample data value along with the key to uniquely identify the record is provided in the output html.
* Html Report is generated based on the Metadata and Data Comparison result.
* The Source and Target Data taken for comparison can also be configured to be saved as .csv files

# Appendix:

Sample Table Key Config File:



**Point of Contact:**

**Developers:** Bhargav M A Gorpade, Bhuvanendra Kumar Jothiramalingam

**Project Manager:** Aparna Mohandas